反復丸めアルゴリズム

福永 拓郎

国立情報学研究所
JST, ERATO, 河原林巨大グラフプロジェクト
takuro@nii.ac.jp

2013 年 7 月 25 日 組合せ最適化セミナー

反復丸めとは？

- 線形計画問題を利用した近似アルゴリズムの設計手法の一つ
- Jain のシュタイナーネットワーク 2 近似アルゴリズム (1998) 以降, 様々な問題で新しいアルゴリズムが反復丸め法に基づいて発見されてきた
- 組合せ最適化の古典的な結果の多くも反復丸め法で証明できることが分かってきた
- 抑えなければいけない細かいことがいくつかある
- アイデアはそんなに難しくないので，基本を理解すればあとは簡単
- まだまだたくさん適用できる問題や，新しいアイデアを盛り込む余地が残されているはず

今日のながれ

1. 反復丸めアルゴリズムの枠組み
   - 様々な線形計画丸めアルゴリズム
   - 反復丸めアルゴリズムの特徴
   - 階数補題: LP 端点解の特徴付け
2. 端点解の疎性を利用したアルゴリズム
   - 全域木問題
   - 次数制約付き全域木
   - 多目的全域木問題
3. LP 解の更新を利用したアルゴリズム
   - シュタイナー木問題

今日のながれ

1. 反復丸めアルゴリズムの枠組み
   - 様々な線形計画丸めアルゴリズム
   - 反復丸めアルゴリズムの特徴
   - 階数補題: LP 端点解の特徴付け
2. 端点解の疎性を利用したアルゴリズム
   - 全域木問題
   - 次数制約付き全域木
   - 多目的全域木問題
3. LP 解の更新を利用したアルゴリズム
   - シュタイナー木問題

近似アルゴリズム

近似アルゴリズム (最小化問題の場合)

- OPT: 最適解の目的関数値
- ALG: アルゴリズムの出力解の目的関数値
- α: 1 以上の実数

どのような問題例に対してもアルゴリズムの出力解が

$$\text{ALG} \leq \alpha \times \text{OPT}$$

を満たすならば，そのアルゴリズムを α **近似アルゴリズム**と呼び，
α を**近似比**と呼ぶ．

近似比を証明するには

1. OPT の適当な下界値を見つけ
2. ALG $\leq \alpha \times$ (下界値) を証明する

線形計画丸め

組合せ最適化問題

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \in \{0,1\}^n \end{array}$$

緩和 ➜

線形計画問題

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \in [0,1]^n \end{array}$$

LP $\leq$ OPT だから， ALG $\leq \alpha \times$ LP となるアルゴリズムを設計すれば良い． ただし，

線形計画丸め

組合せ最適化問題

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \{0, 1\}^n \end{aligned}$$

緩和 ➜

線形計画問題

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in [0, 1]^n \end{aligned}$$

LP $\leq$ OPT だから， ALG $\leq \alpha \times$ LP となるアルゴリズムを設計すれば良い． ただし，

線形計画問題の整数性ギャップ

$$\text{整数性ギャップ} = \sup \frac{\text{OPT}}{\text{LP}}$$

整数性ギャップ $\geq \beta$ ➜ その線形計画問題を使う限り β より良い近似比は達成できない

いろいろな線形計画丸めアルゴリズムの枠組み

- 整数性，半整数性，$1/k$-整数性
- 主双対法
- 双対フィッティング
- ランダム丸め
- 反復丸め

半整数性

Vertex Cover 問題

- **入力:** 無向グラフ $G = (V, E)$, 点コスト $c : V \to \mathbb{R}_+$
- **Vertex Cover** $U \subseteq V$: $\forall e \in E$ が U のどれかの点に接続している
- **出力:** 最小コスト Vertex Cover

Vertex Cover LP

$$
\begin{array}{lll}
\min & \sum_{v \in V} c(v)x(v) & \\
\text{s.t.} & x(u) + x(v) \geq 1, & uv \in E \\
& x(v) \geq 0, & v \in V
\end{array}
$$

半整数性

Vertex Cover 問題

- **入力:** 無向グラフ $G = (V, E)$, 点コスト $c : V \to \mathbb{R}_+$
- **Vertex Cover** $U \subseteq V$: $\forall e \in E$ が U のどれかの点に接続している
- **出力:** 最小コスト Vertex Cover

Vertex Cover LP

$$
\begin{array}{lll}
\min & \sum_{v \in V} c(v)x(v) & \\
\text{s.t.} & x(u) + x(v) \geq 1, & uv \in E \\
 & x(v) \geq 0, & v \in V
\end{array}
$$

定理

$x(v) \in \{0, 1/2, 1\}$ $(\forall v \in V)$ を満たす LP の最適解 x が存在する.

Vertex Cover の 2 近似アルゴリズム

アルゴリズム

1. LP の最適解 x を計算する
2. $\{v \in V \mid x(v) \geq 1/2\}$ を出力

Vertex Cover の 2 近似アルゴリズム

アルゴリズム

1. LP の最適解 x を計算する
2. $\{v \in V \mid x(v) \geq 1/2\}$ を出力

定理

上のアルゴリズムが出力する解は実行可能かつ $2 \times$ LP 以下のコストを持つ． ➜ 2 近似アルゴリズム

Vertex Cover の 2 近似アルゴリズム

アルゴリズム

1. LP の最適解 x を計算する
2. $\{v \in V \mid x(v) \geq 1/2\}$ を出力

定理

上のアルゴリズムが出力する解は実行可能かつ $2 \times$ LP 以下のコストを持つ. ➜ 2 近似アルゴリズム

欠点: こんないい性質は滅多に成り立たない.

主双対法

主問題

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

=

双対問題

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y \leq c \\ & y \in \mathbb{R}^m_+ \end{aligned}$$

主双対法

主問題

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

=

双対問題

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y \leq c \\ & y \in \mathbb{R}^m_+ \end{aligned}$$

方針

主問題の整数実行可能解 x と双対問題の実行可能解 y のうち
$c^T x \leq \alpha b^T y$ を満たすものを構築する

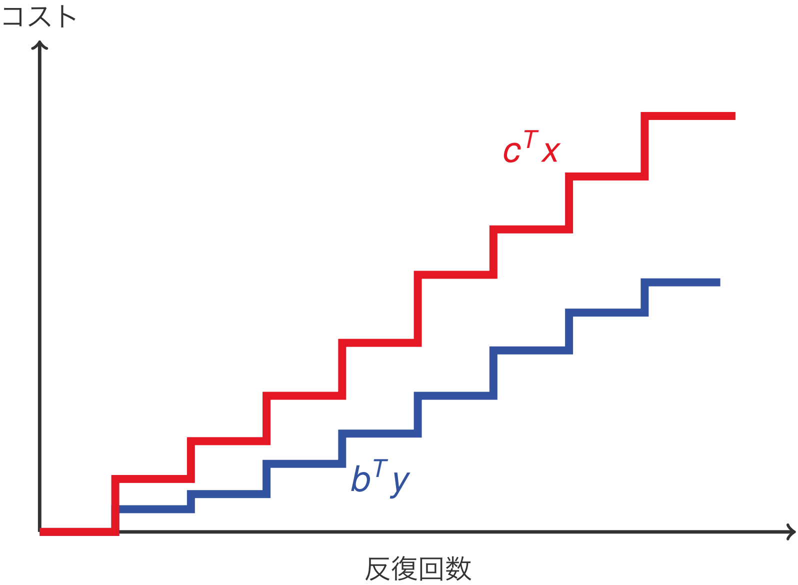
⬇

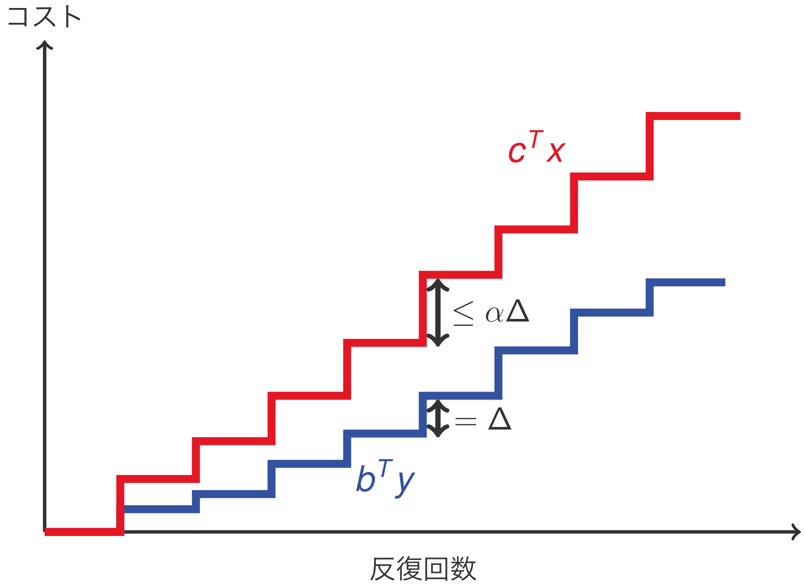
$$\mathrm{ALG} = c^T x \leq \alpha b^T y \leq \alpha\ \mathrm{OPT}$$

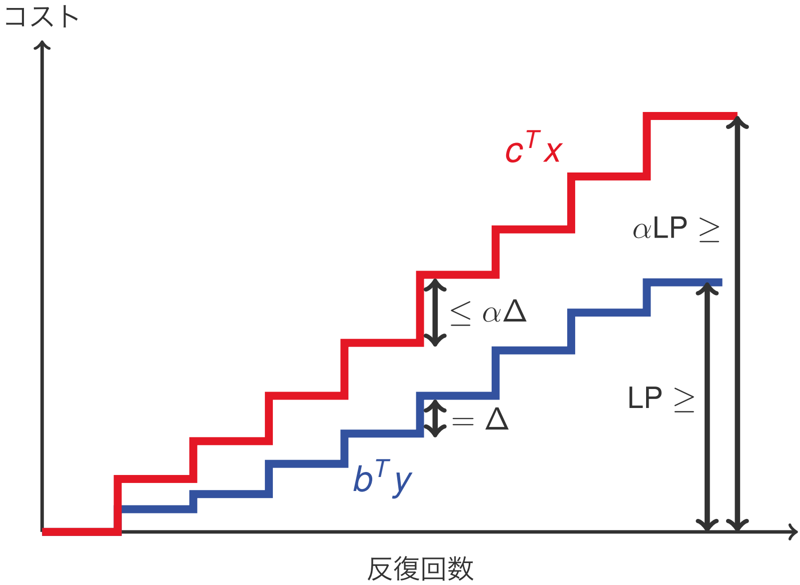
典型的な主双対法アルゴリズム

アルゴリズム

1. $x \leftarrow 0$, $y \leftarrow 0$
2. 以下を満たすように x と y を増加させる
   - y は双対問題の実行可能解
   - ($c^T x$ の増加量) $\leq \alpha \times$ ($b^T y$ の増加量)
3. x が主問題の実行可能解になったら x を出力して終了.
   そうでなかったら 2 に戻る.


コスト
$c^T x$
$b^T y$
反復回数


コスト
$c^T x$
$\leq \alpha\Delta$
$= \Delta$
$b^T y$
反復回数


コスト
$c^T x$
$\alpha\mathrm{LP} \geq$
$\leq \alpha\Delta$
$= \Delta$
$\mathrm{LP} \geq$
$b^T y$
反復回数

反復丸め法

$1/k$ 整数性

- すべての変数 $\geq 1/k$ もしくは $= 0$ ➜ k 近似
- 適用できる問題がほとんどない

反復丸め法

$1/k$ 整数性

- すべての変数 $\geq 1/k$ もしくは $= 0$ ➜ k 近似
- 適用できる問題がほとんどない

反復丸め法

- 変数のうちどれか一つが $\geq 1/\alpha$ もしくは $= 0$ ➜ α 近似
- 上の性質が再帰的に成り立つ必要がある

反復丸め法

$1/k$ 整数性

- すべての変数 $\geq 1/k$ もしくは $= 0$ ➜ k 近似
- 適用できる問題がほとんどない

反復丸め法

- 変数のうちどれか一つが $\geq 1/\alpha$ もしくは $= 0$ ➜ α 近似
- 上の性質が再帰的に成り立つ必要がある

反復丸めアルゴリズム

1. LP の最適解 x を計算する
2. 0 の値をもつ変数 x_i を 0 に固定する
3. $1/\alpha$ 以上の値をもつ変数 x_i を $\lceil x_i \rceil$ に固定する
4. すべての変数が固定されたらそれを出力して終了．そうでなかったら 1 に戻る

再帰的な構造を持つ LP

例: Edge Cover 問題

- 入力: 無向グラフ $G = (V, E)$, 辺コスト $c : E \to \mathbb{R}_+$
- Edge Cover $F \subseteq E$: $\forall v \in V$ に接続する辺が少なくとも一つ F に含まれている
- 出力: 最小コスト Edge Cover

再帰的な構造を持つ LP

例: Edge Cover 問題

- 入力: 無向グラフ $G = (V, E)$, 辺コスト $c : E \to \mathbb{R}_+$
- Edge Cover $F \subseteq E$: $\forall v \in V$ に接続する辺が少なくとも一つ F に含まれている
- 出力: 最小コスト Edge Cover

- $\delta_E(v)$ は E に含まれる辺のうち v に接続しているものの集合を表す
- $E' \subseteq E$ について $x(E')$ は $\sum_{e \in E'} x(e)$ を表す

Edge Cover LP

$$\begin{array}{lll} \min & \sum_{e \in E} c(e)x(e) & \\ \text{s.t.} & x(\delta_E(v)) \geq 1, & v \in V \\ & x(e) \geq 0, & e \in E \end{array}$$

再帰的な構造を持つ LP

例: Edge Cover 問題

- 入力: 無向グラフ $G = (V, E)$, 辺コスト $c : E \to \mathbb{R}_+$
- Edge Cover $F \subseteq E$: $\forall v \in V$ に接続する辺が少なくとも一つ F に含まれている
- 出力: 最小コスト Edge Cover

- $\delta_E(v)$ は E に含まれる辺のうち v に接続しているものの集合を表す
- $E' \subseteq E$ について $x(E')$ は $\sum_{e \in E'} x(e)$ を表す

Edge Cover LP

$$\begin{array}{lll} \min & \sum_{e \in E} c(e)x(e) & \\ \text{s.t.} & x(\delta_E(v)) \geq 1, & v \in V \\ & x(e) \geq 0, & e \in E \end{array}$$

➜ ダメ

F をそれまでに変数を 1 に固定した辺の集合として

再帰的な構造を持つ Edge Cover LP 1

$$
\begin{array}{lll}
\min & \sum_{e \in E \setminus F} c(e)x(e) & \\
\text{s.t.} & x(\delta_{E \setminus F}(v)) \geq 1 - |\delta_F(v)|, & v \in V \\
& x(e) \geq 0, & e \in E \setminus F
\end{array}
$$

もしくは, $x(e)$ を固定したら e をグラフから取り除くということにして

再帰的な構造を持つ Edge Cover LP 2

$$
\begin{array}{lll}
\min & \sum_{e \in E} c(e)x(e) & \\
\text{s.t.} & x(\delta_E(v)) \geq 1, & v \in B\ (\subseteq V) \\
& x(e) \geq 0, & e \in E
\end{array}
$$

ポイント:

- $\forall G, c, F$ (もしくは B) から定義される LP に対して，丸められる変数を持つ最適解が存在することを示す
- 必要となる再帰性は，アルゴリズムでの丸め方に依存する

ただし

Vetex Cover の場合は

- 変数を 1 に固定（点を Vertex Cover に選ぶ）
  ➜ 選んだ点と接続する辺すべてをグラフから取り除く

- 変数を 0 に固定（点を Vertex Cover に選ばない）
  ➜ 選んだ点とその隣接する点，それらに接続する辺を全てグラフから取り除く

とできるので，以下の LP のままでよい．

Vertex Cover LP

$$\begin{array}{lll} \text{min} & \sum_{v \in V} c(v)x(v) & \\ \text{s.t.} & x(u) + x(v) \geq 1, & uv \in E \\ & x(v) \geq 0, & v \in V \end{array}$$

このように，問題自身が再帰性を持つ場合は，LP でそのことを考慮する必要は無い．

反復丸めアルゴリズム

Edge Cover に対するアルゴリズム

1. $F \leftarrow \emptyset$
2. LP の最適解 x を計算する
3. $x(e) = 0$ を満たす辺 e をグラフから取り除く
4. $x(e) \geq 1/\alpha$ を満たす辺 e について $F \leftarrow F \cup \{e\}$ とする
5. $E \setminus F \neq \emptyset$ であれば 2 に戻る．そうでなければ F を出力して終了

定理

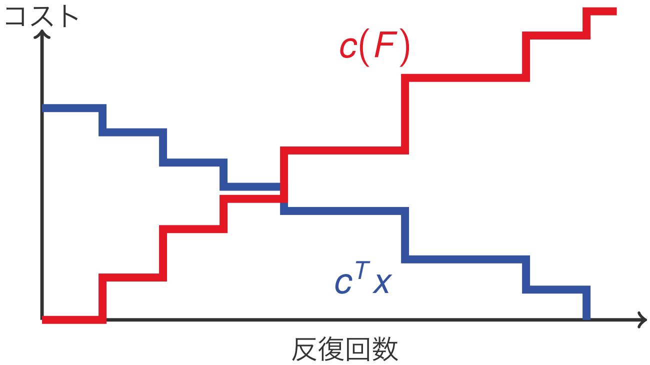
$x(e) = 0$ または $x(e) \geq 1/\alpha$ を満たす辺 e が必ず存在するならば，上のアルゴリズムは α 近似アルゴリズムである．

あとは，なるべく小さい α で上の条件が成り立つことを示せば良い

近似比の証明（方針）

各反復ごとに以下が成り立つことを示す.

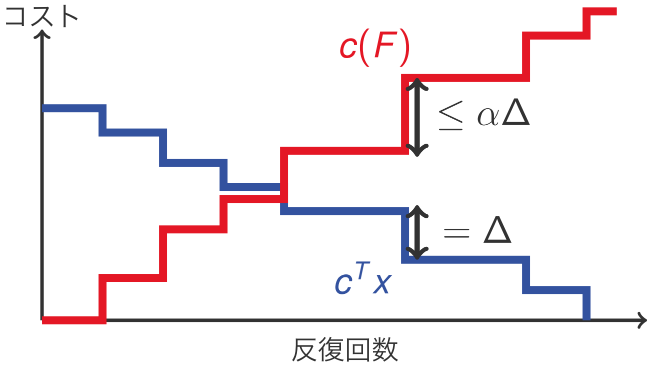
(暫定解 F のコストの増加量) $\leq \alpha \times$ (LP の最適値の減少量)



近似比の証明（方針）

各反復ごとに以下が成り立つことを示す.

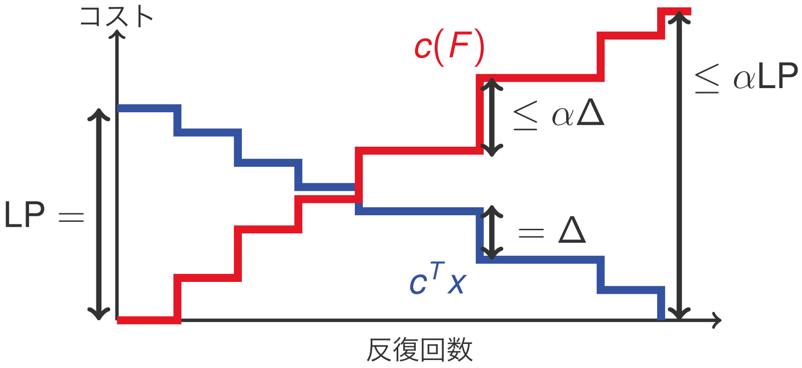
(暫定解 F のコストの増加量) $\leq \alpha \times$ (LP の最適値の減少量)



近似比の証明（方針）

各反復ごとに以下が成り立つことを示す．

(暫定解 F のコストの増加量) $\leq \alpha \times$ (LP の最適値の減少量)



近似比の証明（詳細）

$x(e) = 0$ を満たす辺 e が存在するとき

- 暫定解 F のコストの増加量 = 0
- LP の最適値の減少量 = 0

$x(e) \geq 1/\alpha$ を満たす辺 e が存在するとき

- 暫定解 F のコストの増加量 $= c(e) \leq \alpha c(e)x(e)$
  ($x(e) \geq 1/\alpha$ だから)
- LP の最適値の減少量 $\geq c(e)x(e)$
  (更新前の LP の最適解から変数 $x(e)$ を取り除いたものは更新後の LP の実行可能解だから)

コメント: 仮に $x(e) < 1/\alpha$ であっても，LP の最適値が充分小さくなることが証明できるのであれば丸めても α 近似解を計算できる

反復丸め法のまとめ

やること

1. 再帰的な LP を定義する
2. LP の最適解が，0 or 充分大きい値をとる変数を持つことを示す (もしくは LP の最適値が充分小さくなるような丸め方を考える)

- 以上の議論は最小化問題で，制約が変数を下から抑えるようなもののみの場合の話．最大化問題や，制約が上下から挟むようなものの場合は工夫が必要
- 2 番目のことを示すための常套手段が LP の端点解の性質の利用

主双対法 v.s. 反復丸め法

	主双対法		反復丸め法
○	組合せ的なアルゴリズム (LP の汎用アルゴリズムを使わなくてもよい)	×	LP を解く必要がある
×	アルゴリズムの設計が複雑	○	アルゴリズム自体は単純 (LP を解くところ以外は)
○	対応できる問題の形は柔軟	×	上下両方から抑えるような制約があると工夫なしには適用が難しい
○	枠組み自体は柔軟で多くの種類のアルゴリズムがある	×	バリエーションが少ない (単に歴史が浅いだけかも)

高い値をとる変数の存在性

いいたいこと

- α: 十分小さい実数
- $P := \{x \in \mathbb{R}^n \mid Ax \geq b, x \geq 0\}$
- $\min_{x \in P} c^T x$ の最適解のうち以下を満たすものが存在する

$$\exists \text{変数 } x_i\text{: } x_i = 0 \text{ もしくは } x_i \geq 1/\alpha \qquad (\bigstar)$$

補足

- $x = 0$ が実行可能解でないかぎり，$(\bigstar) \Rightarrow$「$\exists$ 変数 x_i: $x_i \geq 1/\alpha$」
- 変数 x_i が $(\bigstar)$ の条件を満たすかは $\min\{c^T x : x \in P\} = \min\{c^T x : x \in P, x_i \geq 1/\alpha\}$ が成立するかをチェックすれば判定できる．つまり，$(\bigstar)$ を満たす最適解の存在性さえ証明できれば，実際にそれを計算できなくてもよい

端点最適解

定義

次の条件を満たす $y \in \mathbb{R}^n$ が存在しないような実行可能解 x を端点解と呼ぶ.

- $y \neq 0$
- $x + y \in P$ かつ $x - y \in P$

端点最適解

定義

次の条件を満たす $y \in \mathbb{R}^n$ が存在しないような実行可能解 x を端点解と呼ぶ.

- $y \neq 0$
- $x + y \in P$ かつ $x - y \in P$

定理

$\min_{x \in P} c^T x$ が有限のとき,

- 必ず端点解であるような最適解が存在する
- 最適解を多項式時間で計算できる
  $\Rightarrow$ 端点最適解を多項式時間で計算できる

よって，任意の端点解が (★) を満たすことを示せばよい.

階数補題

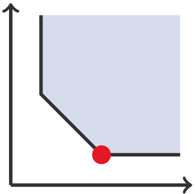
定義

- 制約 $A_j x \geq b_j$ が (実行可能解 x^* について) タイトである
  $\Leftrightarrow A_j x^* = b_j$ が成立
- $\{A_j x \geq b_j : j \in J\}$ が線形独立 $\Leftrightarrow$ ベクトル A_j ($j \in J$) が線形独立

階数補題

- x: $\min\{c^T x : Ax \geq b, x \geq 0\}$ の端点最適解
- $J : \{A_j x \geq b_j : j \in J\}$ が線形独立でタイトな制約の極大族

とする．このとき，$x > 0$ ならば **x の次元 = $|J|$**.



階数補題の証明

観察

- $x \in P$
- $A^=$: x についてタイトな制約に対応する行からなる A の部分行列
- $A^=_x$: 変数 $x_i > 0$ に対応する列からなる $A^=$ の部分行列

$$x \text{ が端点解} \Leftrightarrow A^=_x \text{ の列は線形独立}$$

$\Rightarrow$)

1. $I = \{i \in [n] : x_i > 0\}$
2. $A^=_x$ の列は線形独立じゃないと仮定する. つまり
   $\exists y \in R^{|I|} : y \neq 0, A^=_x y = 0$
3. $i \in [n] \setminus I$ について $y_i = 0$ となるよう y を n 次元ベクトルに拡張
4. 十分小さい $\epsilon > 0$ について $x + \epsilon y, x - \epsilon y \in P$.
   つまり x は端点解ではない.

階数補題の証明（続き）

階数補題 (再掲)

- x: $\min\{c^T x : Ax \geq b, x \geq 0\}$ の端点最適解
- J : $\{A_j x \geq b_j : j \in J\}$ が線形独立でタイトな制約の極大族

とする．このとき，$x > 0$ ならば **x の次元 = $|J|$**.

証明

1. $x > 0$ より，$A^{=} = A^{=}_{x}$
2. 補題より $A^{=}$ はフルランク．つまり rank($A^{=}$) = (x の次元)
3. $\mathrm{rank}(A^{=}) = |J|$

階数補題の意味

- $x > 0$ とすると，タイトな制約は「$x(\delta_E(v)) \geq 1, v \in V$」のみ
- 階数定理より，
  $|E| =$ 変数の数 $= |J| \leq |V|$
- 一方，
  $x(E) = \sum_{v \in V} x(\delta_E(v))/2 \geq |V|/2$

Edge Cover LP

$$\begin{array}{ll} \min & \sum_{e \in E} c(e)x(e) \\ \text{s.t.} & x(\delta_E(v)) \geq 1, \;\; v \in V \\ & x(e) \geq 0, \;\; e \in E \end{array}$$

階数補題の意味

- $x > 0$ とすると，タイトな制約は「$x(\delta_E(v)) \geq 1,\ v \in V$」のみ
- 階数定理より，
  $|E| =$ 変数の数 $= |J| \leq |V|$
- 一方，
  $x(E) = \sum_{v \in V} x(\delta_E(v))/2 \geq |V|/2$

Edge Cover LP

$$
\begin{array}{ll}
\text{min} & \sum_{e \in E} c(e)x(e) \\
\text{s.t.} & x(\delta_E(v)) \geq 1,\ \ v \in V \\
 & x(e) \geq 0,\ \ e \in E
\end{array}
$$

よって，

$$\max_{e \in E} x(e) \geq \frac{x(E)}{|E|} \geq \frac{|V|/2}{|V|} = \frac{1}{2}$$

階数補題の意味

- $x > 0$ とすると，タイトな制約は「$x(\delta_E(v)) \geq 1, v \in V$」のみ
- 階数定理より，
  $|E| =$ 変数の数 $= |J| \leq |V|$
- 一方，
  $x(E) = \sum_{v \in V} x(\delta_E(v))/2 \geq |V|/2$

Edge Cover LP

$$
\begin{array}{ll}
\min & \sum_{e \in E} c(e)x(e) \\
\text{s.t.} & x(\delta_E(v)) \geq 1, \ v \in V \\
 & x(e) \geq 0, \ e \in E
\end{array}
$$

よって，

$$\max_{e \in E} x(e) \geq \frac{x(E)}{|E|} \geq \frac{|V|/2}{|V|} = \frac{1}{2}$$

定理

Edge Cover LP の整数性ギャップ ≤ 2. 反復丸め法は，Edge Cover の 2 近似アルゴリズムを与える.

階数補題の意味

- $x > 0$ とすると，タイトな制約は「$x(\delta_E(v)) \geq 1,\ v \in V$」のみ
- 階数定理より，$|E| =$ 変数の数 $= |J| \leq |V|$
- 一方，$x(E) = \sum_{v \in V} x(\delta_E(v))/2 \geq |V|/2$

Edge Cover LP

$$\begin{array}{ll} \min & \sum_{e \in E} c(e)x(e) \\ \text{s.t.} & x(\delta_E(v)) \geq 1,\ v \in V \\ & x(e) \geq 0,\ e \in E \end{array}$$

グラフが疎である

よって，

$$\max_{e \in E} x(e) \geq \frac{x(E)}{|E|} \geq \frac{|V|/2}{|V|} = \frac{1}{2}$$

定理

Edge Cover LP の整数性ギャップ ≤ 2. 反復丸め法は，Edge Cover の 2 近似アルゴリズムを与える.

補足

- 最初に整数値を取る変数をすべて丸めてしまえば，残った辺の本数は高々 $|V|$ ➜ 反復回数 = $O(|V|)$
- Edge Cover の LP 緩和としては，以下の LP が整数性ギャップ = 1 であることが知られている

$$
\begin{array}{lll}
\min & \sum_{e\in E} c(e)x(e) & \\
\text{s.t.} & x(E[U]) + x(\delta(U)) \geq \lceil |U|/2 \rceil, & U \subseteq V, |U| \text{ is odd} \\
 & 0 \leq x(e) \leq 1, & e \in E
\end{array}
$$

ただし，

- $E[U]$: U に両端点が含まれる辺の集合
- $\delta(U)$: U に片方の端点が含まれる辺の集合

今日のながれ

1. 反復丸めアルゴリズムの枠組み
   - 様々な線形計画丸めアルゴリズム
   - 反復丸めアルゴリズムの特徴
   - 階数補題: LP 端点解の特徴付け

2. 端点解の疎性を利用したアルゴリズム
   - 全域木問題
   - 次数制約付き全域木
   - 多目的全域木問題

3. LP 解の更新を利用したアルゴリズム
   - シュタイナー木問題

全域木問題

全域木問題

- **入力:** 無向グラフ $G = (V, E)$, 辺コスト $c : E \to \mathbb{R}_+$
- **全域木** $T \subseteq E$: 全ての点を連結にする閉路の無い部分グラフ
- **出力:** 最小コスト全域木

Covering LP

$$
\begin{array}{ll}
\min & \sum_{e \in E} c(e)x(e) \\
\text{s.t.} & x(\delta(U)) \geq 1, \quad U \subset V, U \neq \emptyset \\
 & x(e) \geq 0, \quad e \in E
\end{array}
$$

再帰性

- 辺 e を木に加える ➜ 縮約
- 辺 e を木に加えない ➜ 除去

全域木問題

全域木問題

- **入力:** 無向グラフ $G = (V, E)$, 辺コスト $c : E \to \mathbb{R}_+$
- **全域木** $T \subseteq E$: 全ての点を連結にする閉路の無い部分グラフ
- **出力:** 最小コスト全域木

Covering LP

$$
\begin{array}{ll}
\min & \sum_{e \in E} c(e)x(e) \\
\text{s.t.} & x(\delta(U)) \geq 1, \quad U \subset V, U \neq \emptyset \\
& x(e) \geq 0, \quad e \in E
\end{array}
$$

再帰性

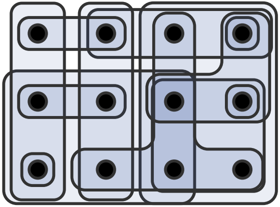
- 辺 e を木に加える ➜ 縮約
- 辺 e を木に加えない ➜ 除去

定理

上の LP は $x(e) = 0$ または $x(e) \geq 1/2$ を満たす辺 e が存在するような最適解を持つ．（よって，整数性ギャップ ≤ 2)

タイトな制約の構造

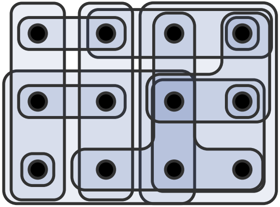
制約の数 $= 2^{|V|} - 2$ ➜ タイトな制約の数 $\leq 2^{|V|} - 2$ ➜ ?



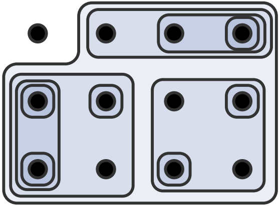
タイト制約

タイトな制約の構造

制約の数 $= 2^{|V|} - 2$ ➜ タイトな制約の数 $\leq 2^{|V|} - 2$ ➜ ?



タイトな制約

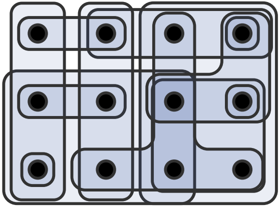


線形独立なタイトな制約

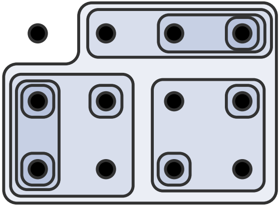
➜ ラミナー

タイトな制約の構造

制約の数 = $2^{|V|} - 2$ ➜ タイトな制約の数 $\leq 2^{|V|} - 2$ ➜ ?



タイト制約



線形独立なタイト制約

➜ ラミナー

定義

- $X, Y \in 2^V$ が**交差している** $\Leftrightarrow X \cap Y, X \setminus Y, Y \setminus X \neq \emptyset$
- $X, Y \in 2^V$ が**交差していない** $\Leftrightarrow X \cap Y = \emptyset$ or $X \subseteq Y$ or $Y \subseteq X$
- $\mathcal{F} \subseteq 2^V$ が**ラミナー** $\Leftrightarrow \forall X, Y \in \mathcal{F}$ が交差していない

線形独立なタイト制約の構造

定理

$\forall e \in E$ について $x(e) > 0$ とする．このとき，x について線形独立でタイトな制約の極大族のうち，**ラミナーなものが存在する**．

よって，(線形独立でタイトな制約の数) $\leq 2|V| - 1$

$$\rightarrow \max_{e \in E} x(e) \geq \frac{x(E)}{|E|} \geq \frac{|V|/2}{2|V|-1} = \frac{1}{4} + O\left(\frac{1}{|V|}\right)$$

劣モジュラ性

- $\chi_U \in \{0,1\}^E$: $\delta(U)$ の生成ベクトル. つまり,

$$\chi_U(e) = \begin{cases} 1 & \text{if } e \in \delta(U) \\ 0 & \text{if } e \notin \delta(U) \end{cases}$$

- **劣モジュラ性** $\forall X, Y \subseteq V : \chi_X + \chi_Y \geq \chi_{X\cap Y} + \chi_{X\cup Y}$
- これより, $\forall x : E \to \mathbb{R}_+$ に対して

  $\forall X, Y \subseteq V : x(\delta(X)) + x(\delta(Y)) \geq x(\delta(X \cap Y)) + x(\delta(X \cup Y))$

補題

$X \cap Y \neq \emptyset, X \cup Y \neq V$ である $X, Y \subset V$ について制約がタイトならば,

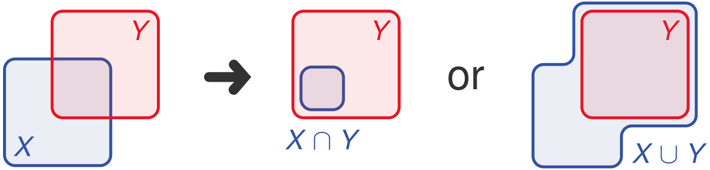
- $X \cap Y, X \cup Y$ についての制約もタイト
- $\chi_X + \chi_Y = \chi_{X\cap Y} + \chi_{X\cup Y}$

証明: $1 + 1 = x(\delta(X)) + x(\delta(Y)) \geq x(\delta(X \cap Y)) + x(\delta(X \cup Y)) \geq 1 + 1$

ラミナー性の証明

- $|U| \leq |V| - |U|$ であるような $U \subset V$ についてのカット制約のみ考える
- 線形独立でラミナーなタイト制約の族の中で極大なものを $\mathcal{L}$ とする
- 任意のタイト制約 $X \notin \mathcal{L}$ について，$\mathcal{L} \cup \{X\}$ が線形独立だと仮定する
- $\mathcal{L}$ の極大性より $\mathcal{L} \cup \{X\}$ はラミナーじゃないはずなので，X と交差している $Y \in \mathcal{L}$ が存在

- 前々ページの補題より，$X \cap Y, X \cup Y$ 両方の制約もタイト
- $\mathcal{L} \cup \{X \cap Y\}$ もしくは $\mathcal{L} \cup \{X \cup Y\}$ のどちらかは線形独立
  なぜなら $\chi_X = \chi_{X \cap Y} + \chi_{X \cup Y} - \chi_Y$ より，$\chi_{X \cap Y}, \chi_{X \cup Y}$ の両方が $\mathcal{L}$ に従属していれば χ_X も従属しており，$\mathcal{L} \cup \{X\}$ が線形独立であることに矛盾する
- $\mathcal{L} \cup \{X \cap Y\}, \mathcal{L} \cup \{X \cup Y\}$ どちらもラミナーなので，$\mathcal{L}$ の極大性に矛盾する

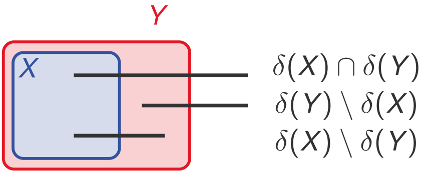


➜ $\mathcal{L}$ は線形独立なタイト制約族の中で極大

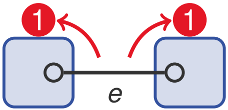
さらに

ラミナー性と線形独立性からいえること

- 点 v から定義される集合族 $\mathcal{F}_v = \{X \in \mathcal{F} \mid v \in X\}$ の中で極小のものはただ一つ
  (極大線形独立タイト制約 $\mathcal{F}$ のラミナー性より)
- $X, Y \in \mathcal{F}, X \subset Y : |\delta(X) \setminus \delta(Y)| + |\delta(Y) \setminus \delta(X)| > 0$
  (χ_X と χ_Y の線形独立性より)
- さらに $|\delta(X) \setminus \delta(Y)| + |\delta(Y) \setminus \delta(X)| \geq 2$
  ($x(\delta(X)) = x(\delta(Y)) = 1$ より)



トークンを使った数え上げ



仮定

- $\forall e = uv \in E : 0 < x(e) < 1/3$
- e はトークンを 2 つ持っている
- $e = uv$ は $\mathcal{F}_u$ の中で極小な制約にトークンを一つ，$\mathcal{F}_v$ の中で極小な制約にトークンをもう一つ与える

トークンの再分配

- $\forall X \in \mathcal{F}$ は少なくとも 2 つトークンを集めることができる
- 極大な $X \in \mathcal{F}$ は 4 つ以上のトークンを集めることができる

➜ $2|\mathcal{F}| + 2 \leq$ トークンの数 $\leq 2|E|$

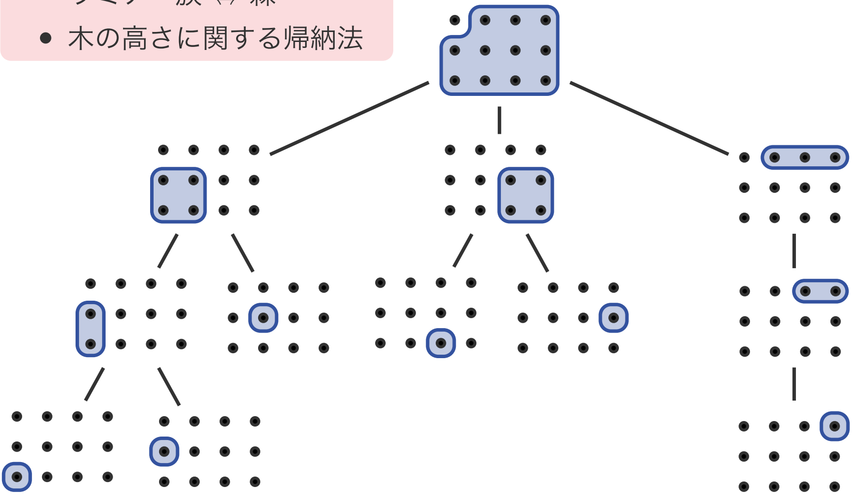
➜ $|E| = |\mathcal{F}|$ に 矛盾

定理

Covering LP は，$x(e) = 0$ もしくは $x(e) \geq 1/3$ を満たす辺 e が存在するような最適解 x を持つ．

トークンの再分配方法

- ラミナー族 ⇔ 森
- 木の高さに関する帰納法



帰納法の枠組み

木の高さ=1

- $\forall e \in E : 0 < x(e) < 1/3,\ x(\delta(X)) = 1 \Rightarrow |\delta(X)| \geq 4$
- つまり $X \in \mathcal{F}$ は 4 つ以上のトークンがもらえる

木の高さ ≥ 2

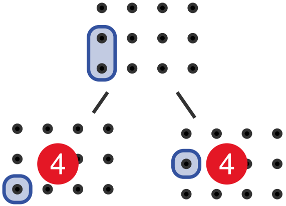
- $X \in \mathcal{F}$ を極大なものとする
- X の子供を頂点とする部分木それぞれについて帰納法を適用
  - X の子供はそれぞれ 4 つのトークンを受け取っている
  - X の子供より下にいる点集合はそれぞれ 2 つのトークンを受け取っている

ケース 1　X の子供が 2 人以上の場合
ケース 2　X の子供が 1 人のみの場合

木の高さ ≥ 2 の場合

ケース 1　X の子供が 2 人以上の場合

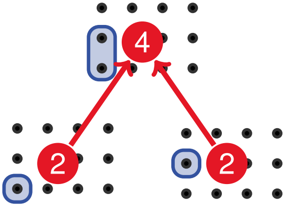
子供それぞれが 2 つずつトークンを X に分け与える



木の高さ ≥ 2 の場合

ケース 1　X の子供が 2 人以上の場合

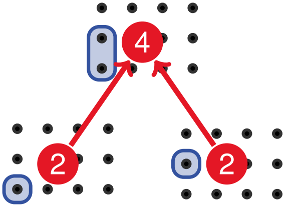
子供それぞれが 2 つずつトークンを X に分け与える



木の高さ ≥ 2 の場合

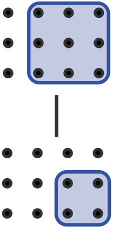
ケース 1　X の子供が 2 人以上の場合

子供それぞれが 2 つずつトークンを X に分け与える



ケース 2　X の子供が 1 人のみの場合

- 子供 Y は X にトークン 2 個分け与える
- X は $e \in (\delta_E(X) \setminus \delta_E(Y)) \cup (\delta_E(Y) \setminus \delta_E(X))$ からトークンを 1 つずつもらう

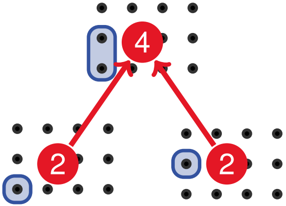


この証明は $0 < x(e) < 1/2$ の場合にも拡張可能

木の高さ ≥ 2 の場合

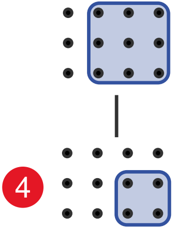
ケース 1　X の子供が 2 人以上の場合

子供それぞれが 2 つずつトークンを X に分け与える



ケース 2　X の子供が 1 人のみの場合

- 子供 Y は X にトークン 2 個分け与える
- X は $e \in (\delta_E(X) \setminus \delta_E(Y)) \cup (\delta_E(Y) \setminus \delta_E(X))$ からトークンを 1 つずつもらう

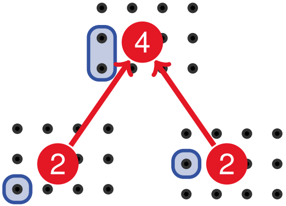


この証明は $0 < x(e) < 1/2$ の場合にも拡張可能

木の高さ ≥ 2 の場合

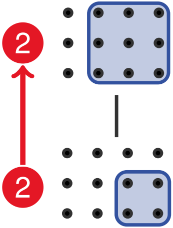
ケース 1　X の子供が 2 人以上の場合

子供それぞれが 2 つずつトークンを X に分け与える



ケース 2　X の子供が 1 人のみの場合

- 子供 Y は X にトークン 2 個分け与える
- X は $e \in (\delta_E(X) \setminus \delta_E(Y)) \cup (\delta_E(Y) \setminus \delta_E(X))$ からトークンを 1 つずつもらう

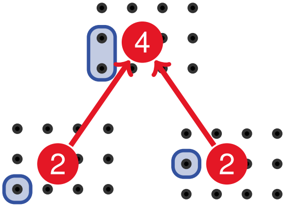


この証明は $0 < x(e) < 1/2$ の場合にも拡張可能

木の高さ ≥ 2 の場合

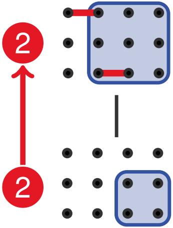
ケース 1　X の子供が 2 人以上の場合

子供それぞれが 2 つずつトークンを X に分け与える



ケース 2　X の子供が 1 人のみの場合

- 子供 Y は X にトークン 2 個分け与える
- X は $e \in (\delta_E(X) \setminus \delta_E(Y)) \cup (\delta_E(Y) \setminus \delta_E(X))$ からトークンを 1 つずつもらう

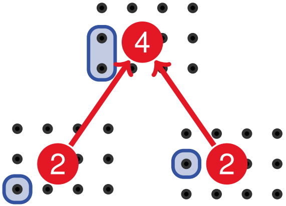


この証明は $0 < x(e) < 1/2$ の場合にも拡張可能

木の高さ ≥ 2 の場合

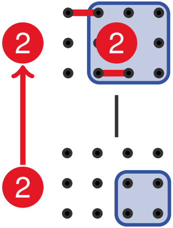
ケース 1　X の子供が 2 人以上の場合

子供それぞれが 2 つずつトークンを X に分け与える



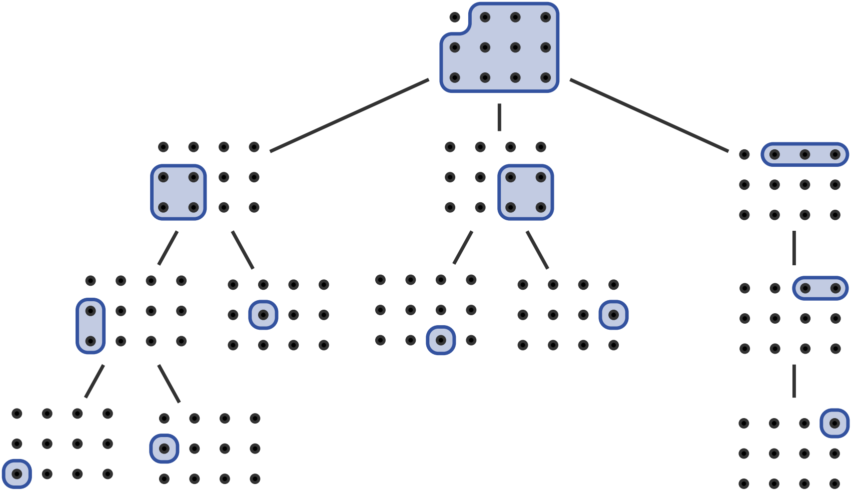
ケース 2　X の子供が 1 人のみの場合

- 子供 Y は X にトークン 2 個分け与える
- X は $e \in (\delta_E(X) \setminus \delta_E(Y)) \cup (\delta_E(Y) \setminus \delta_E(X))$ からトークンを 1 つずつもらう

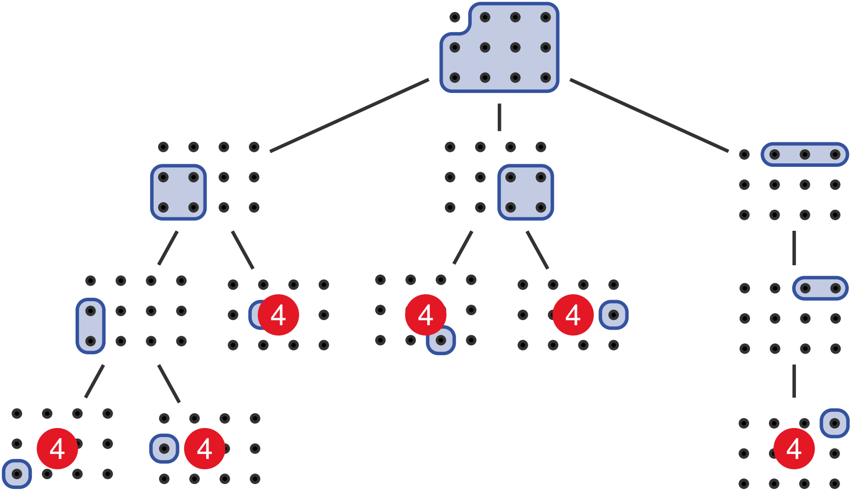


この証明は $0 < x(e) < 1/2$ の場合にも拡張可能

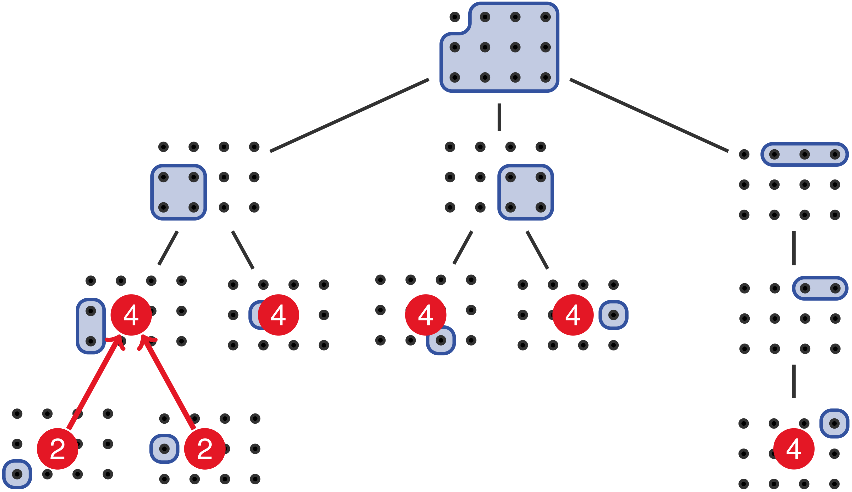
帰納法の全体像



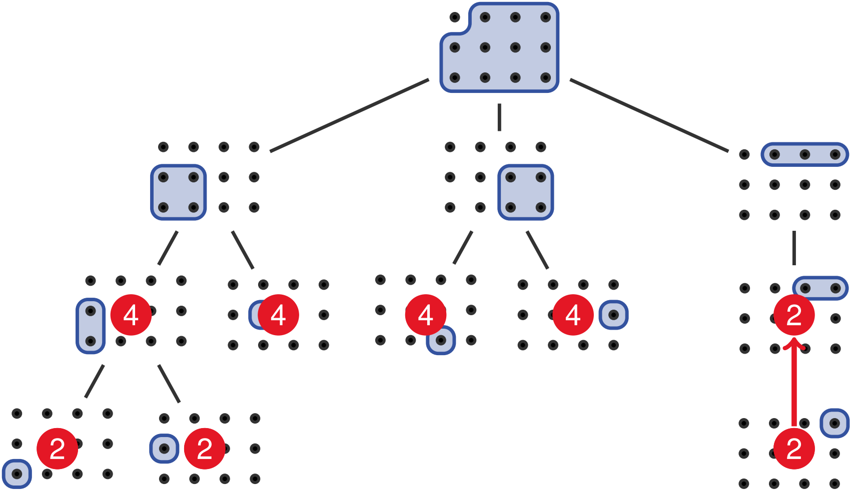
帰納法の全体像



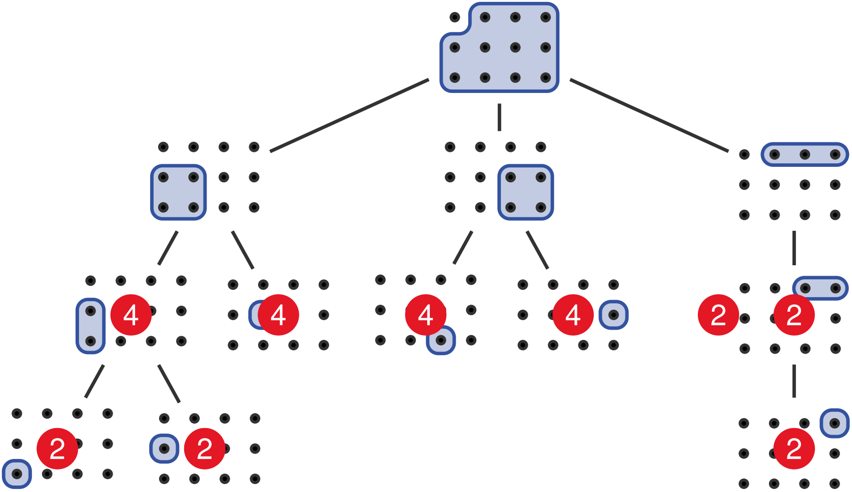
帰納法の全体像



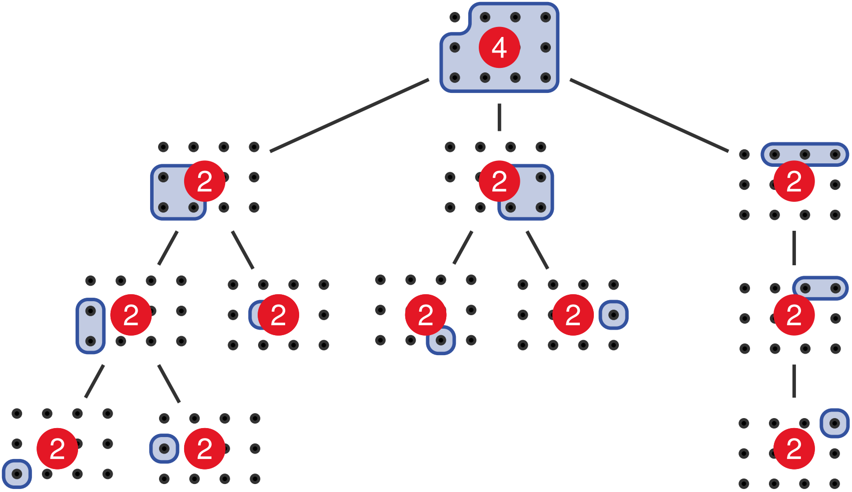
帰納法の全体像



帰納法の全体像



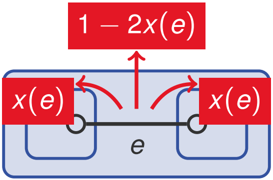
帰納法の全体像



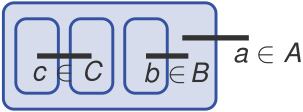
有理数トークンを使った数え上げ

トークンの配り方

- $\forall e \in E : 0 < x(e) < 1/2$ と仮定する
- e はトークンを 1 つ持っている
- $e = uv$ は
  - $\mathcal{F}_u$ の中で極小なものにトークン $x(e) > 0$
  - $\mathcal{F}_v$ の中で極小なものにトークン $x(e) > 0$
  - $\{X \in \mathcal{F} \mid u, v \in X\}$ の中で極小なものにトークン $1 - 2x(e) > 0$ を与える



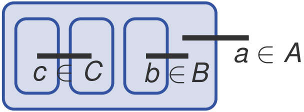
$X \in \mathcal{F}$ が受け取るトークンの量



$$
\begin{aligned}
\textbf{X の受け取るトークン} &= \sum_{a \in A} x(a) + \sum_{b \in B} (1 - x(b)) + \sum_{c \in C} (1 - 2x(c)) \\
&= x(A) - x(B) - 2x(C) + |B| + |C| \\
&= x(\delta(X)) - \sum_{i=1}^{k} x(\delta(Y_i)) + |B| + |C| \\
&= 1 - k + |B| + |C|
\end{aligned}
$$

(ただし，$Y_1, Y_2, \ldots, Y_k$ は X の子供)

$X \in \mathcal{F}$ が受け取るトークンの量



$$
\begin{aligned}
X\text{ の受け取るトークン} &= \underbrace{\sum_{a \in A} x(a) + \sum_{b \in B} (1 - x(b)) + \sum_{c \in C} (1 - 2x(c))}_{>0} \\
&= x(A) - x(B) - 2x(C) + |B| + |C| \\
&= x(\delta(X)) - \sum_{i=1}^{k} x(\delta(Y_i)) + |B| + |C| \\
&= \underbrace{1 - k + |B| + |C|}_{\text{整数}} \quad \boxed{\geq 1}
\end{aligned}
$$

(ただし， $Y_1, Y_2, \ldots, Y_k$ は X の子供)

有理数トークン分配のまとめ

- 各 $e \in E$ は高々 1 のトークンを配る
- 各 $X \in \mathcal{F}$ は 1 以上のトークンを受け取る
- 極大な $X \in \mathcal{F}$ について，$e \in \delta(X)$ が配るトークン $1 - 2x(e) > 0$ は誰も受け取らずに余っている

$$|\mathcal{F}| \leq (\text{トークンの総量}) < |E| \qquad \text{➜矛盾}$$

定理

Covering LP は，$x(e) = 0$ もしくは $x(e) \geq 1/2$ を満たす辺 e が存在するような最適解 x を持つ．

Subtour Elimination LP

Subtour Elimination LP

$$
\begin{array}{lll}
\min & \sum_{e \in E} c(e) x(e) & \\
\text{s.t.} & x(E[U]) \leq |U| - 1, & U \subset V \\
& x(E) = |V| - 1 & \\
& x(e) \geq 0, & e \in E
\end{array}
$$

定理

$x(e) > 0, \forall e \in E$ であるような Subtour Elimination LP の端点解について線形独立でタイトな制約の極大族 $\mathcal{F} \subseteq 2^V$ のうち，**ラミナー**なものが存在する．

Subtour Elimination LP

Subtour Elimination LP

$$
\begin{array}{lll}
\min & \sum_{e\in E} c(e)x(e) & \\
\text{s.t.} & x(E[U]) \le |U| - 1, & U \subset V \\
& x(E) = |V| - 1 & \\
& x(e) \ge 0, & e \in E
\end{array}
$$

定理

$x(e) > 0, \forall e \in E$ であるような Subtour Elimination LP の端点解について線形独立でタイトな制約の極大族 $\mathcal{F} \subseteq 2^V$ のうち，**ラミナー**なものが存在する．

$\mathcal{F}$ は singleton を持たないラミナー族 ➜ $|\mathcal{F}| \le |V| - 1$

$$
\max_{e\in E} x(e) \ge \frac{x(E)}{|E|} = \frac{|V|-1}{|\mathcal{F}|} \ge \frac{|V|-1}{|V|-1} = 1
$$

Subtour Elimination LP

Subtour Elimination LP

$$\begin{array}{lll} \min & \sum_{e \in E} c(e)x(e) & \\ \text{s.t.} & x(E[U]) \leq |U| - 1, & U \subset V \\ & x(E) = |V| - 1 & \\ & x(e) \geq 0, & e \in E \end{array}$$

定理

$x(e) > 0, \forall e \in E$ であるような Subtour Elimination LP の端点解について線形独立でタイトな制約の極大族 $\mathcal{F} \subseteq 2^V$ のうち，**ラミナー**なものが存在する．

$\mathcal{F}$ は singleton を持たないラミナー族 ➜ $|\mathcal{F}| \leq |V| - 1$

$$\max_{e \in E} x(e) \geq \frac{x(E)}{|E|} = \frac{|V| - 1}{|\mathcal{F}|} \geq \frac{|V| - 1}{|V| - 1} = 1$$

定理

Subtour Elimination LP は，$x(e) \in \{0, 1\}$ を満たす辺 e が存在するような最適解 x を持つ．

次数最小全域木

次数最小全域木問題

- **入力:** 無向グラフ $G = (V, E)$
- $d_T(v)$: 全域木 T での点 v の次数
- **出力:** $\max_{v \in V} d_T(v)$ を最小化する全域木 T

- 「OPT= 2 $\Leftrightarrow$ G がハミルトン閉路を持つ」
  ➜ 次数最小全域木問題は NP 困難
- Fürer, Raghavachari '94: 最大次数 OPT+1 以下の全域木を計算する多項式時間アルゴリズム

次数制約付き最小全域木

次数制約付き最小全域木問題

- **入力:** 無向グラフ $G = (V, E)$, 次数制約 $b : V \to \mathbb{Z}_+$, 辺コスト $c : E \to \mathbb{R}_+$
- **次数制約:** $d_T(v) \leq b(v), \forall v \in V$
- **出力:** 次数制約を満たし $c(T)$ を最小化する全域木 T

- 実行可能解を見つけることも NP 困難
- Goemans '06: 実行可能解が存在するならば, $d_T(v) \leq b(v) + 2$ $(\forall v \in V)$ かつ $c(T) \leq \mathrm{OPT}$ であるような全域木 T を計算するアルゴリズム
- Singh, Lau '07: 実行可能解が存在するならば, $d_T(v) \leq b(v) + 1$ $(\forall v \in V)$ かつ $c(T) \leq \mathrm{OPT}$ であるような全域木 T を計算するアルゴリズム

次数制約付き最小全域木問題

Degree bounded LP

$$
\begin{array}{lll}
\min & \sum_{e \in E} c(e)x(e) & \\
\text{s.t.} & x(E[U]) \le |U| - 1, & U \subset V \\
& x(E) = |V| - 1 & \\
& x(\delta(v)) \le b(v), & v \in V \\
& x(e) \ge 0, & e \in E
\end{array}
$$

定理

$\forall e \in E$ について $x(e) > 0$ とする．このとき，x について線形独立でタイトな制約の極大族のうち，

- ラミナー族 $\mathcal{F} \subseteq \{U \subseteq V \mid |U| \ge 2\}$ 内によって定義される Subtour Elimination 制約
- $W \subseteq V$ 内の点に関する次数制約

から成るものが存在する．

端点解の性質 1

端点解 x が $\forall e \in E$ について $x(e) > 0$ を満たすとする.

$\mathcal{F}$ のラミナー性より

$|\mathcal{F}| \leq |V| - 1$ かつ $|W| \leq |V|$ ➜ $|E| = |\mathcal{F}| + |W| \leq 2|V| - 1$

端点解の性質 1

端点解 x が $\forall e \in E$ について $x(e) > 0$ を満たすとする.

$\mathcal{F}$ のラミナー性より

$$|\mathcal{F}| \leq |V| - 1 \text{ かつ } |W| \leq |V| \rightarrow |E| = |\mathcal{F}| + |W| \leq 2|V| - 1$$

$$\Downarrow$$

$$\forall X \subseteq V \text{ について } |E[X]| \leq 2|X| - 3$$

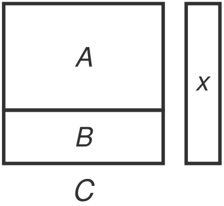
ここでは $|E[X]| \leq 2|X| - 1$ を示す

$|E[X]| \leq 2|X| - 1$ の証明

- $x(E[U]) = |U| - 1, U \in \mathcal{F}$ を $Ax = a$,
  $x(\delta(v)) = b(v), v \in W$ を $Bx = b$ と書く

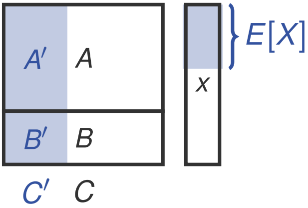
- $C = \begin{bmatrix} A \\ B \end{bmatrix}$ は $|E| \times |E|$ の正方行列で，rank$(C) = |E|$

- A, B, C から $E[X]$ の辺に対応する列だけを取ってきた部分行列を A', B', C' とする．rank$(C') = |E[X]|$.

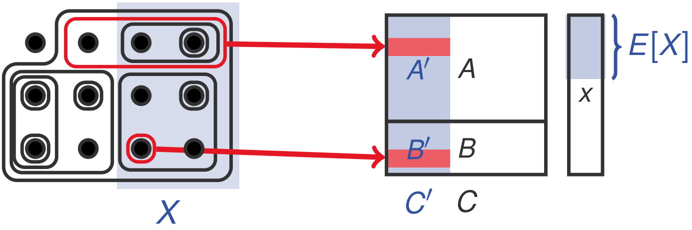


$|E[X]| \leq 2|X| - 1$ の証明

- $x(E[U]) = |U| - 1, U \in \mathcal{F}$ を $Ax = a$,
  $x(\delta(v)) = b(v), v \in W$ を $Bx = b$ と書く
- $C = \begin{bmatrix} A \\ B \end{bmatrix}$ は $|E| \times |E|$ の正方行列で，rank$(C) = |E|$
- A, B, C から $E[X]$ の辺に対応する列だけを取ってきた部分行列を A', B', C' とする．rank$(C') = |E[X]|$.



$|E[X]| \le 2|X| - 1$ の証明



- A' の行のうち非ゼロ要素をもつものは，$U \in \mathcal{F}, |U \cap X| \ge 2$ に対応する Subtour Elimination 制約．それらは X 上のラミナー族 ➜ そのような行の数は重複を除くと高々 $|X| - 1$
- B' の行のうち非ゼロ要素をもつものは，$v \in W \cap X$ に対応する次数制約 ➜ そのような行の数 $\le |X|$
- よって，$\mathrm{rank}(C') \le 2|X| - 1$

グラフの疎性 → グラフ向き付け → マトロイド交差

定理（次数制約グラフ向き付け，Hakimi）

$\forall X \subseteq V$ について $|E[X]| \leq 2|X| - 1$ ならば，全ての点の入次数 ≤ 2 となるよう無向グラフ G を向き付けできる．

定理（マトロイド交差，Edmonds）

以下の LP に整数最適解 ($= |\delta^+(v)| \leq b(v)$ となる無向全域木のなかでコスト最小) が存在し，多項式時間で計算できる．

$$
\begin{array}{lll}
\min & \sum_{e \in E} c(e)x(e) & \\
\text{s.t.} & x(E[U]) \leq |U| - 1 & U \subset V \\
& x(E) = |V| - 1 & \\
& x(\delta^+(v)) \leq b(v) & v \in V \\
& 0 \leq x(e) \leq 1 & e \in E
\end{array}
$$

+2 アルゴリズム

アルゴリズム

1. Degree bounded LP の端点最適解 x を計算する
2. $x(e) = 0$ を満たす辺 e をグラフから除去
3. 入次数 ≤ 2 となるようグラフを向き付け
4. $|\delta^+(v)| \leq b(v)$ となる無向全域木のなかでコスト最小のものを出力

$d_T(v) \leq b(v) + 2$ の証明

$$d_T(v) = |\delta^+(v)| + |\delta^-(v)| \leq b(v) + 2$$

端点解の性質 2

再帰的な Degree bounded LP

$$
\begin{array}{lll}
\min & \sum_{e \in E} c(e)x(e) & \\
\text{s.t.} & x(E[U]) \leq |U| - 1 & U \subset V \\
& x(E) = |V| - 1 & \\
& x(\delta(v)) \leq b(v) & v \in B \quad (\subseteq V) \\
& 0 \leq x(e) \leq 1 & e \in E
\end{array}
$$

定理

端点解 x が $\forall e \in E$ について $x(e) > 0$ を満たし，かつ $B \neq \emptyset$ とする．このとき，$d_E(v) \leq b(v) + 1$ を満たす点 $v \in B$ が存在する．

+1 アルゴリズム

アルゴリズム

1. $B \neq \emptyset$ であるかぎり，以下を繰り返す
   - LP の端点最適解 x を計算する
   - $x(e) = 0$ を満たす辺 e をグラフから除去
   - $d_E(v) \leq b(v) + 1$ を満たす $v \in B$ を B から除去
2. 最小コスト全域木を出力

出力解 T は以下を満たす

$$d_T(v) \leq d_E(v) \leq b(v) + 1$$

性質 2 の証明

- 以下を仮定
  - $\forall e \in E : x(e) > 0$
  - $\forall v \in B : d_E(v) \geq b(v) + 2$

- 線形独立でタイトな制約の極大族は以下のものから成る.
  - ラミナー族 $\mathcal{F} \subseteq \{U \subseteq V \mid |U| \geq 2\}$ によって定義される Subtour Elimination 制約
  - $W \subseteq B$ 内の点に関する次数制約

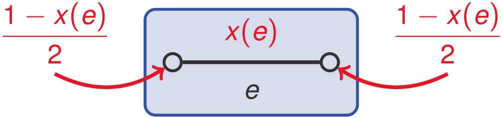
ケース 1: $W = \emptyset$ ➜ 整数端点解が存在 ➜ $d_E(v) \leq b(v)$

ケース 2: $W \neq \emptyset$ ➜ トークンを使った数え上げ

ケース 2 $W \neq \emptyset$

トークンの配り方　$\forall e \in E$ は

- e の両端点を含む極小な集合 $U \in \mathcal{F}$ にトークン $x(e)$ を与える
- e の端点のうち W に含まれるものにトークン $\dfrac{1 - x(e)}{2}$ を与える



$$e \text{ が配るトークンの量} \leq x(e) + 2 \times \frac{1 - x(e)}{2} = 1$$

$\mathcal{F} \cup W$ が受け取るトークンの量

$U \in \mathcal{F}$ の受け取るトークンの量

$C_1, C_2, \ldots, C_k \in \mathcal{F}$ を U の子供とすると

$$U\text{ のトークン} = x(E[U]) - \sum_{i=1}^{k} x(E[C_i]) = |U| - 1 - \sum_{i=1}^{k}(|C_i| - 1)$$

これは整数，かつ > 0 (線形独立性より) ➜ U のトークン ≥ 1.

$v \in W$ の受け取るトークンの量

$$v\text{ のトークン} = \sum_{e \in \delta(v)} \frac{1 - x(e)}{2} = \frac{d_E(v) - b(v)}{2} \geq 1$$

ここまでで，$|E| \leq$ トークンの総量 $\leq |\mathcal{F}| + |W|$.
余っているトークンがあることが証明できれば矛盾が示せる.

余っているトークン

- $V \notin \mathcal{F}$ ➜ $\forall U \in \mathcal{F}$ について $e \notin E[U]$ となる辺 e が存在 (グラフは連結だから) ➜ e の $\mathcal{F}$ へのトークンは誰も受け取っていない
- $W \neq B$ ➜ $v \in B \setminus W$ に接続している辺のトークンが余っている
- $v \in V \setminus B$, $e \in \delta(v)$, $x(e) < 1$ を満たす点 v と辺 e が存在する
  ➜ e のトークン $(1 - x(e))/2 > 0$ が余っている

それ以外のとき

$$
\begin{aligned}
2\chi(E[V]) &= \sum_{v \in V} \chi(\delta(v)) \\
&= \sum_{v \in B} \chi(\delta(v)) + \sum_{v \in V \setminus B} \chi(\delta(v)) \\
&= \sum_{v \in W} \chi(\delta(v)) + \sum_{v \in V \setminus B} \sum_{uv \in \delta(v)} \chi(\{u, v\})
\end{aligned}
$$

余っているトークン

- $V \notin \mathcal{F}$ ➜ $\forall U \in \mathcal{F}$ について $e \notin E[U]$ となる辺 e が存在 (グラフは連結だから) ➜ e の $\mathcal{F}$ へのトークンは誰も受け取っていない
- $W \neq B$ ➜ $v \in B \setminus W$ に接続している辺のトークンが余っている
- $v \in V \setminus B,\ e \in \delta(v),\ x(e) < 1$ を満たす点 v と辺 e が存在する
  ➜ e のトークン $(1 - x(e))/2 > 0$ が余っている

それ以外のとき

$$
\begin{aligned}
2\chi(E[V]) &= \sum_{v\in V} \chi(\delta(v)) \\
&= \sum_{v\in B} \chi(\delta(v)) + \sum_{v\in V\setminus B} \chi(\delta(v)) \\
&= \sum_{v\in W} \chi(\delta(v)) + \sum_{v\in V\setminus B} \sum_{uv\in\delta(v)} \boxed{\chi(\{u,v\})}
\end{aligned}
$$

$\in \mathrm{span}(\chi(\delta(U)), U \in \mathcal{F})$

$x(E[\{u,v\}]) \le 1$
はタイト

よって線形独立性に矛盾

多目的全域木問題

多目的全域木問題

- **入力:** 無向グラフ $G = (V, E)$, 辺コスト $c_1, c_2, \ldots, c_k : E \to \mathbb{R}_+$
- **出力:** $c_1(T), c_2(T), \ldots, c_k(T)$ を同時に最小化する全域木 T

多目的全域木問題

多目的全域木問題

- **入力:** 無向グラフ $G = (V, E)$, 辺コスト $c_1, c_2, \ldots, c_k : E \to \mathbb{R}_+$
- **出力:** $c_1(T), c_2(T), \ldots, c_k(T)$ を同時に最小化する全域木 T

定理 [Grandoni, Ravi, Singh '09]

G が

$$c_i(T^*) \le b_i, \quad i = 1, 2, \ldots, k$$

を満たす全域木 T^* を持つならば，任意の $\epsilon > 0$ に対して

- $c_1(T) \le b_1$
- $c_i(T) \le b_i + \epsilon$, $i = 2, 3, \ldots, k$

を満たす全域木 T を計算する多項式時間アルゴリズムが存在する．
ただし，k と ϵ は定数とする．

アルゴリズム

前処理

- $c_i(e) \geq \epsilon b_i/(k-1)$ であるような辺 e のうち T^* に含まれるものは高々 $(k-1)/\epsilon$ 本
- そのような辺のうち T^* に含まれるものを推測するには, $O(|E|^{(k-1)/\epsilon})$ 通りの組合せを試せばよい
- すべての $i \geq 2$ について推測するには, $O(|E|^{(k-1)^2/\epsilon})$ 通りの組合せを試せばよい

仮定

$\forall e \in E, \forall i \geq 2$ について $c_i(e) < \epsilon b_i/(k-1)$

線形計画問題

Multi Obj LP

$$
\begin{array}{lll}
\min & \sum_{e\in E} c_1(e)x(e) & \\
\text{s.t.} & x(E[U]) \leq |U| - 1, & U \subset V \\
 & x(E) = |V| - 1 & \\
 & \sum_{e\in E} c_i(e)x(e) \leq b_i, & i = 2, 3, \ldots, k \\
 & x(e) \geq 0, & e \in E
\end{array}
$$

制約の構造

定理

Multi Obj LP の端点解 x が $\forall e \in E$ について $x(e) > 0$ とする．
このとき， $|E| \leq |V| + k - 2$.

証明: タイトな制約の極大な線形独立族は以下のものから成る．

- ラミナー族 $\mathcal{F} = \{U \subseteq V \mid |U| \geq 2\}$ から定義される Subtour Elimination 制約
- $\sum_{e \in E} c_i(e)x(e) = b_i, i \in I \subseteq \{2, 3, \ldots, k\}$

よって $|E| \leq |\mathcal{F}| + |I| \leq |V| + k - 2$.

アルゴリズムの詳細

1. $E_i = \{c_i(e) \geq \epsilon b_i/(k-1)\}, i \geq 2$ とする.
2. 各 $i = 2, 3, \ldots, k$ について, E_i から高々 $(k-1)/\epsilon$ の辺を選ぶ組合せを列挙する,
3. それぞれの組合せについて以下のことを行い，得られた全域木の中で c_1 コスト最小のものを出力する.
   - 3.1 E_i から組合せに選ばれた辺 e を縮約し，$b_i := b_i - c_i(e)$ と更新する $(i \geq 2)$
   - 3.2 $E_2 \cup E_3 \cup \cdots \cup E_k$ に含まれる辺で組合せに選ばれなかった辺を除去する
   - 3.3 LP の端点解 x を計算し，$x(e) = 0$ となっている辺 e を除去する.
   - 3.4 c_1 コスト最小全域木を計算する.

アルゴリズムの出力解のコスト

$c_1(T) \leq b_1$

- E_i, $i \geq 2$ からの組合せが正しければ，Multi Obj LP $\leq c_1(T^*)$.
- Subtour Elimination LP $\leq$ Multi Obj LP
- アルゴリズムが出力する T は「$c_1(T) \leq$ Subtour Elimination LP」を満たす

$c_i(T) \leq b_i + \epsilon$, $i \geq 2$

- 簡単のため，前処理は無視し，$c_i(e_1) \leq c_i(e_2) \leq \cdots \leq c_i(e_{n+k-2})$ であるとする
- $c_i(e_1) + c_i(e_2) + \cdots + c_i(e_{n-1}) \leq \sum_{e \in E} c_i(e)x(e) \leq b_i$
- $c_i(T) \leq c_i(e_k) + \cdots + c_i(e_{n-1}) + c_i(e_n) + \cdots + c_i(e_{n+k-2})$
  $\leq b_i + (k-1) \times \epsilon b_i/(k-1) = (1+\epsilon)b_i$

今日のながれ

1. 反復丸めアルゴリズムの枠組み
   - 様々な線形計画丸めアルゴリズム
   - 反復丸めアルゴリズムの特徴
   - 階数補題: LP 端点解の特徴付け
2. 端点解の疎性を利用したアルゴリズム
   - 全域木問題
   - 次数制約付き全域木
   - 多目的全域木問題
3. LP 解の更新を利用したアルゴリズム
   - シュタイナー木問題

シュタイナー木問題

シュタイナー木問題

- **入力:** 無向完全グラフ $G = (V, E)$, 端点集合 $R \subseteq V$, メトリック辺コスト $c : E \to \mathbb{R}_+$
- **シュタイナー木** $T \subseteq E$: 端点を連結にする木
- **出力:** コスト最小シュタイナー木

近似比

- 2 ($G[R]$ の最小全域木)
- 1.83 [Zelikovsky '93]
- 1.667 [Prömel, Steger '97]
- 1.644 [Karpinski, Zelikovsky '97]
- 1.598 [Hougardy, Prömel '99]
- 1.55 [Robin, Zelikovsky '00]
- 1.39 [Byrka, Grandoni, Rothvoss, Sanitá '10]

シュタイナー木問題

シュタイナー木問題

- **入力:** 無向完全グラフ $G = (V, E)$, 端点集合 $R \subseteq V$, メトリック辺コスト $c : E \to \mathbb{R}_+$
- **シュタイナー木** $T \subseteq E$: 端点を連結にする木
- **出力:** コスト最小シュタイナー木

近似比

整数性ギャップ

- 2 ($G[R]$ の最小全域木) $\longrightarrow$ 2
- 1.83 [Zelikovsky '93]
- 1.667 [Prömel, Steger '97]
- 1.644 [Karpinski, Zelikovsky '97]
- 1.598 [Hougardy, Prömel '99]
- 1.55 [Robin, Zelikovsky '00]
- 1.39 [Byrka, Grandoni, Rothvoss, Sanitá '10]

シュタイナー木問題

シュタイナー木問題

- **入力:** 無向完全グラフ $G = (V, E)$, 端点集合 $R \subseteq V$, メトリック辺コスト $c : E \to \mathbb{R}_+$
- **シュタイナー木** $T \subseteq E$: 端点を連結にする木
- **出力:** コスト最小シュタイナー木

近似比

整数性ギャップ

- 2 ($G[R]$ の最小全域木) $\longrightarrow$ 2
- 1.83 [Zelikovsky '93]
- 1.667 [Prömel, Steger '97]
- 1.644 [Karpinski, Zelikovsky '97]
- 1.598 [Hougardy, Prömel '99] [Chakrabarty, Könemann, Pritchard '10]
- 1.55 [Robin, Zelikovsky '00]
- 1.39 [Byrka, Grandoni, Rothvoss, Sanitá '10] $\longrightarrow$ 1.55

シュタイナー木問題

シュタイナー木問題

- **入力:** 無向完全グラフ $G = (V, E)$, 端点集合 $R \subseteq V$, メトリック辺コスト $c : E \to \mathbb{R}_+$
- **シュタイナー木** $T \subseteq E$: 端点を連結にする木
- **出力:** コスト最小シュタイナー木

近似比

整数性ギャップ

- 2 ($G[R]$ の最小全域木) → 2
- 1.83 [Zelikovsky '93]
- 1.667 [Prömel, Steger '97]
- 1.644 [Karpinski, Zelikovsky '97]
- 1.598 [Hougardy, Prömel '99] [Chakrabarty, Könemann, Pritchard '10]
- 1.55 [Robin, Zelikovsky '00]
- 1.39 [Byrka, Grandoni, Rothvoss, Sanitá '10] → 1.55
- 1.39 [Goemans, Olver, Rothvoss, Zenklusen '12]

線形計画問題

定義

- $C \subseteq R$ の full component: C を葉として持つコスト最小木
- $c(C) :=$ C の full component のコスト
- $\mathcal{K} := 2^R$
- $(x)^+ := \max\{0, x\}$

Hypergrpahic LP

$$
\begin{array}{lll}
\min & \sum_{C \in \mathcal{K}} c(C)x(C) & \\
\text{s.t.} & \sum_{C \in \mathcal{K}} (|S \cap C| - 1)^+ x(c) \leq |S| - 1, & S \subset R, S \neq \emptyset \\
& \sum_{C \in \mathcal{K}} (|R \cap C| - 1)^+ x(c) = |R| - 1 & \\
& x(C) \geq 0, & C \in \mathcal{K}
\end{array}
$$

任意の定数 $\epsilon > 0$ について, 定数 c_ϵ 以下のサイズの端点集合のみを考えることにより，$(1 + \epsilon)$ 近似解を計算できる

アルゴリズム

仮定

- Hypergraphic LP の最適解が計算できる
- 最適解は常に $\sum_{C \in \mathcal{K}} x(C) = M$ を満たす

アルゴリズム

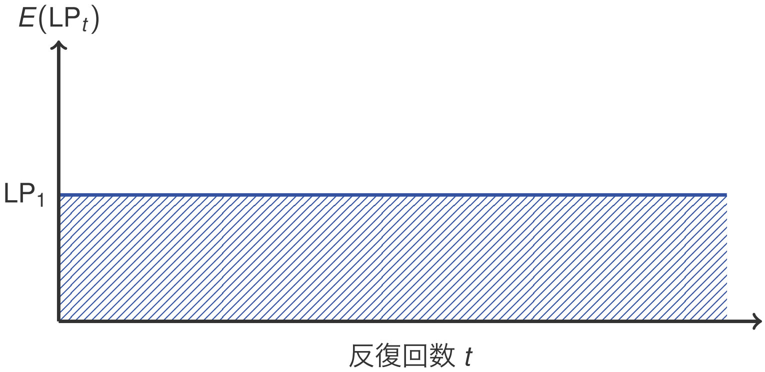
1. 以下を $|R| = 1$ となるまで繰り返す
   1.1 Hypergraphic LP の最適解 x を計算する
   1.2 $C \in \mathcal{K}$ を確率 $x(C)/M$ でサンプルする
   1.3 C の全成分を縮約する
2. 縮約した全成分の和を出力

出力コスト

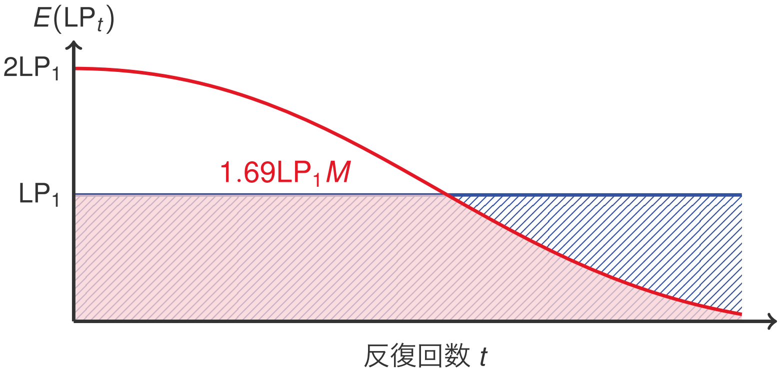
$$c(\text{出力解}) \leq \sum_{t \geq 1} \sum_{C \in \mathcal{K}} \frac{x_t(C)}{M} c_t(C) = \sum_{t \geq 1} \frac{\mathrm{LP}_t}{M}$$

ただし, LP_t は t 反復目の Hypergraphic LP の最適値

近似比の解析



近似比の解析

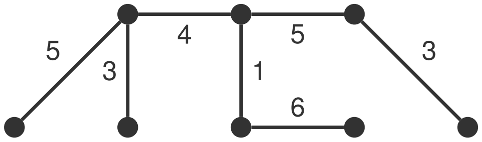


定理

$$E(\mathrm{LP}_t) \leq 2\mathrm{LP}_1 \cdot \left(1 - \frac{1}{M}\right)^{t-1}$$

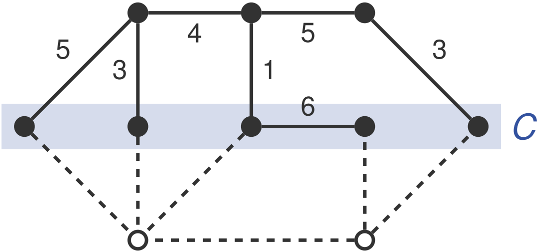
➜ 整数性ギャップ $\leq \sum_{t\geq 1} \min\left\{\frac{1}{M}, \frac{2}{M}\left(1-\frac{1}{M}\right)^t\right\} \leq 1 + \ln(2) \approx 1.69$

ブリッジ



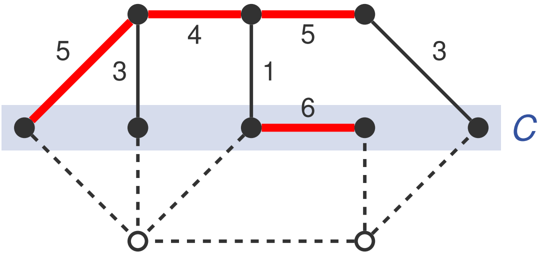
- T: G[R] の全域木

ブリッジ



- T: G[R] の全域木
- $C \in \mathcal{K}$

ブリッジ



- T: G[R] の全域木
- $C \in \mathcal{K}$
- ブリッジ $B_T(C)$: C の full component を T に加えたときに，T から連結性を保ったまま取り除ける辺の集合の中で最大コストのもの

LP_t の期待値

ブリッジ補題

$$\sum_{C \in \mathcal{K}} x(C) c(B_T(C)) \geq c(T)$$

LP_t の期待値

ブリッジ補題

$$\sum_{C\in\mathcal{K}} x(C)c(B_T(C)) \geq c(T)$$

定理 (再掲)

$$E(\mathrm{LP}_t) \leq 2\mathrm{LP}_1 \cdot \left(1-\frac{1}{M}\right)^{t-1}$$

証明

- $G_1[R]$ の最小コスト全域木 T_1 について，$c(T_1) \leq 2\mathrm{LP}_1$
- $C \in \mathcal{K}_t$ の full component を縮約したグラフ $G_{t+1}[R]$ での最小コスト全域木 T_{t+1} について $c(T_{t+1}) \leq c(T_t) - B_{T_t}(C)$ だから,

$$E(\mathrm{LP}_{t+1}) \leq E(c(T_{t+1})) \leq E(c(T_t)) - E(B_{T_t}(C)) \leq \left(1-\frac{1}{M}\right)E(c(T_t))$$

ブリッジ補題の証明

- $\forall u, v \in R$ について $w(uv) := \max\{c(e) \mid e \in (\text{T 上の } uv \text{ パス})\}$ を定義する

観察

C 上の全域木 $S \subseteq C \times C$ で以下を満たすものが存在する.

- $w(S) = c(B_T(C))$
- $\forall uv \in S$ について，T 上の uv パスに含まれる $B_T(C)$ の辺はちょうど一つ

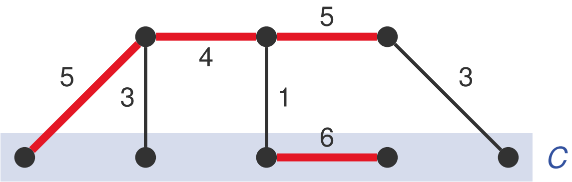
ブリッジ補題の証明

- $\forall u, v \in R$ について $w(uv) := \max\{c(e) \mid e \in (\mathrm{T}$ 上の uv パス$)\}$ を定義する

観察

C 上の全域木 $S \subseteq C \times C$ で以下を満たすものが存在する.

- $w(S) = c(B_T(C))$
- $\forall uv \in S$ について，T 上の uv パスに含まれる $B_T(C)$ の辺はちょうど一つ



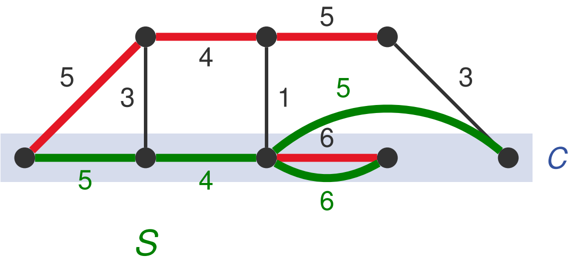
ブリッジ補題の証明

- $\forall u, v \in R$ について $w(uv) := \max\{c(e) \mid e \in (\mathrm{T}$ 上の uv パス$)\}$ を定義する

観察

C 上の全域木 $S \subseteq C \times C$ で以下を満たすものが存在する.

- $w(S) = c(B_T(C))$
- $\forall uv \in S$ について，T 上の uv パスに含まれる $B_T(C)$ の辺はちょうどーつ



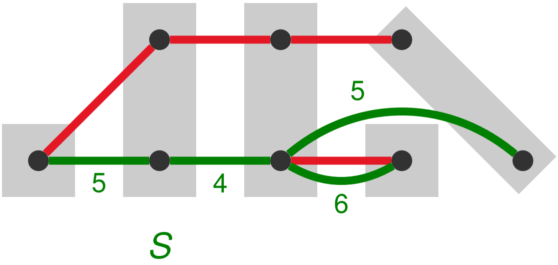
ブリッジ補題の証明

- $\forall u, v \in R$ について $w(uv) := \max\{c(e) \mid e \in (\mathrm{T}$ 上の uv パス$)\}$ を定義する

観察

C 上の全域木 $S \subseteq C \times C$ で以下を満たすものが存在する.

- $w(S) = c(B_T(C))$
- $\forall uv \in S$ について, T 上の uv パスに含まれる $B_T(C)$ の辺はちょうど一つ



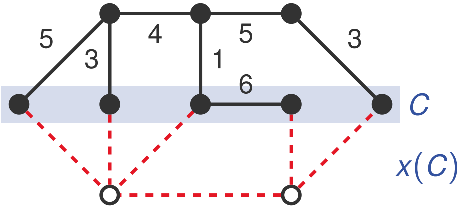
ブリッジ補題の証明

アイデア: ハイパー辺 C を C の全域木に置き換えていく

ブリッジ補題の証明

アイデア: ハイパー辺 C を C の全域木に置き換えていく

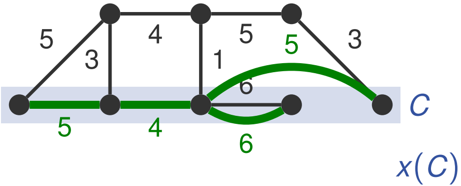
$\forall C \in \mathcal{K}, \forall uv \in S$ について，$y(uv) := x(C)$ を定義する



ブリッジ補題の証明

アイデア: ハイパー辺 C を C の全域木に置き換えていく

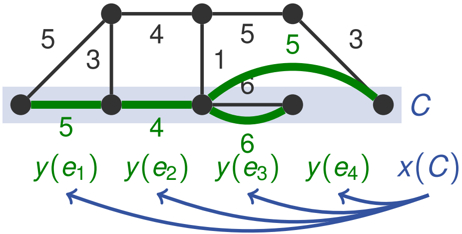
$\forall C \in \mathcal{K}, \forall uv \in S$ について，$y(uv) := x(C)$ を定義する



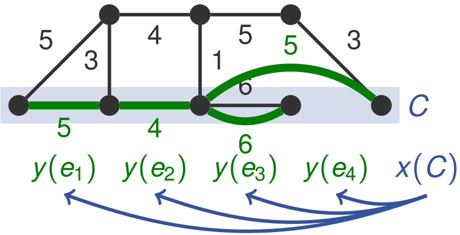
ブリッジ補題の証明

アイデア: ハイパー辺 C を C の全域木に置き換えていく

$\forall C \in \mathcal{K}, \forall uv \in S$ について，$y(uv) := x(C)$ を定義する

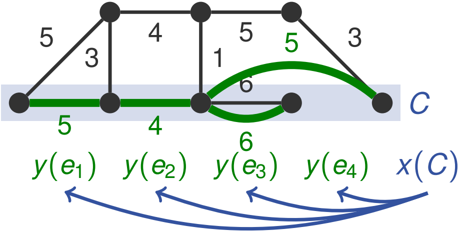


ブリッジ補題の証明



- x が Hypergraphic LP の実行可能解
  ➜ y は $G[R]$ 上の Subtour Elimination LP の実行可能解

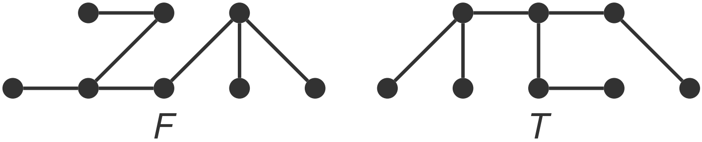
ブリッジ補題の証明



- x が Hypergraphic LP の実行可能解
  ➜ y は $G[R]$ 上の Subtour Elimination LP の実行可能解
- Subtour Elimination LP の整数性より, R 上の全域木 $F \subseteq R \times R$ が存在し, $w(F) \leq \sum_{uv \in R \times R} w(uv)y(uv)$

ブリッジ補題の証明

- w の作り方より $c(T) \leq w(F)$



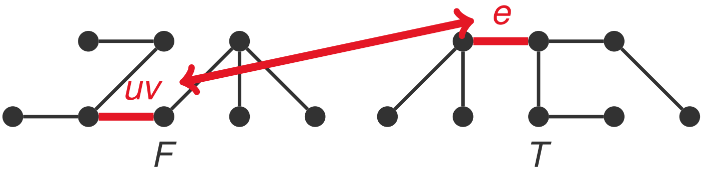
ブリッジ補題の証明

- w の作り方より $c(T) \leq w(F)$

ブリッジ補題の証明

定義:
$w(uv) \quad = \max\{c(e) \mid$
$e \in (T$ 上の uv パス$)\}$

- w の作り方より $c(T) \leq w(F)$

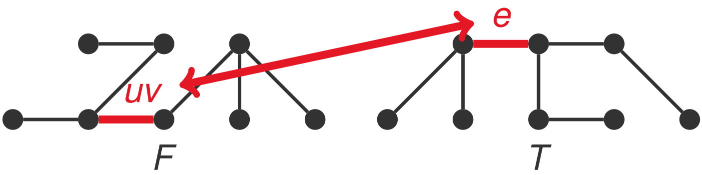


交換可能 & $c(e) \leq w(uv)$

ブリッジ補題の証明

定義:
$w(uv) = \max\{c(e) \mid$
$e \in$ (T 上の uv パス)$\}$

- w の作り方より $c(T) \le w(F)$



交換可能 & $c(e) \le w(uv)$

よって

$$c(T) \le w(F) \le \sum_{uv \in R \times R} w(uv) y(uv) \le \sum_{C \in \mathcal{K}} c(B_T(C)) x(C)$$

反復丸め法とは

- 再帰的な LP
- 階数補題
- 線形独立なタイト制約の良い構造
- 制約の良い構造を利用した数え上げ
- LP 解の更新による近似比の解析

今後の方向性

- まだまだ適用できる問題はたくさんあるはず
- タイト制約がラミナー性のような構造を持たないときにいかに数え上げをするか？
- もっと線形独立性を上手く使えないか？
- 他の LP 丸めの手法との関係は？
- 本当に LP を解かなければならないか？ LP を解くのに汎用アルゴリズムが必要か？

参考文献

反復丸め法について学ぶなら

- Lau, Ravi, Singh. Iterative methods in combinatorial optimization. Cambridge University Press, 2011.

線形計画を用いた近似アルゴリズムについて学ぶなら

- Vazirani. Approximation algorithms. springer, 2001.
- Williamson, Shmoys. The design of approximation algorithms. Cambridge University Press, 2011.

今回取り挙げた論文

- Goemans. Minimum Bounded Degree Spanning Trees. FOCS 2006: 273-282
- Singh, Lau. Approximating minimum bounded degree spanning trees to within one of optimal. STOC 2007: 661-670.
- Grandoni, Ravi, Singh. Iterative Rounding for Multi-Objective Optimization Problems. ESA 2009: 95-106
- Byrka, Grandoni, Rothvoss, Laura Sanitá. An improved LP-based approximation for steiner tree. STOC 2010: 583-592 (J. ACM 60(1): 6 (2013))
- Chan, Lau. On linear and semidefinite programming relaxations for hypergraph matching. SODA 2010: 1500-1511 (Mathematical Programming 135.1-2 (2012): 123-148)